THE HONG KONG POLYTECHNIC UNIVERSITY
HONG KONG COMMUNITY COLLEGE

Subject Title	: Data Structures	**Subject Code**	: CCN2239
Session	: Semester One, 2014/15	**Time**	: 09:30 – 12:30
Date	: 7 December 2014	**Time Allowed**	: 3 Hours
Subject Examiner(s)	: Dr Pat CHAN		

This question paper has a total of **EIGHTEEN** pages (including this covering page).

Instructions to Candidates:

1. There are THREE sections in this paper.
   Section A (40%) – Multiple-choice Questions. Answer ALL questions in this section on the multiple-choice answer sheet provided. Each question carries 1 mark.
   Section B (40%) – Short Questions. Answer any FOUR of the FIVE questions in this section in the answer book provided. Each question carries 10 marks.
   Section C (20%) – Long Questions. Answer any ONE of the TWO questions in this section in the answer book provided. Each question carries 20 marks.
2. Candidates are NOT allowed to retain the multiple-choice answer sheet, the answer book and the examination question paper.
3. Show all your work clearly and neatly. Marks will be deducted for untidy work.
4. Reasonable steps should be shown.
5. All programming code must be written in Java programming language.

Authorised Materials:

	YES	NO
CALCULATOR	[ ]	[✓]
SPECIFICALLY PERMITTED ITEMS	[ ]	[✓]

DO NOT TURN OVER THE PAGE UNTIL YOU ARE TOLD TO DO SO

P

Section B (40%) – Short Questions

Answer any <u>FOUR</u> of the FIVE questions in this section in the answer book provided. Each question carries 10 marks. If more than <u>FOUR</u> questions are answered, only the first <u>FOUR</u> questions answered will be marked.

<u>Question B1</u>

Given the following methods in the stack class:

```
boolean isEmpty();      // true if stack is empty, otherwise false
object pop();           // delete item on top of the stack & return
                        // the reference of it
void push (object O); // push O on the top of the stack
```

(a) Write a code segment to transfer the elements from stack *S1* to stack *S2* so that the elements on *S2* will be in the same order as on *S1* by using one additional stack. (6 marks)

(b) What is the difference between a stack and a queue? (4 marks)

<u>Question B2</u>

Given the following classes:

```
 class ListNode {
   private int value;    //data value
   public ListNode next; //next element of list, or null if last
   public ListNode(int v) { value = v; }
   public int value() { return value; }
 }

class LinkedList
{
   // data members
   public ListNode head; //pointing to the first ListNode
   public int size;      //size of the linked list
   // methods of LinkedList come here
   ...
}
```

(a) Write a recursive delete method for singly-linked lists with integer data that deletes the first occurrence of a given integer from the list and returns the resulting list. (6 marks)

(b) Write a method for attaching a singly linked-list to the end of another singly linked list. (4 marks)

Question B3

Consider the following list of words:

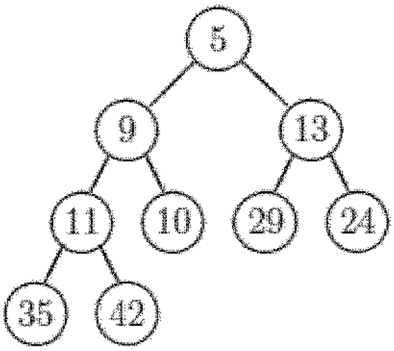
```
apple, tree, car, dog, yellow, frog, gun, harp
```

(a) Alphabetize the above list using a merge sort. Show each step in detail. (6 marks)

(b) Consider an initially empty binary search tree (BST). Place each of the above words in the order given above into the BST. (Use alphabetical order to make your comparisons.) Draw the completed binary search tree. (4 marks)

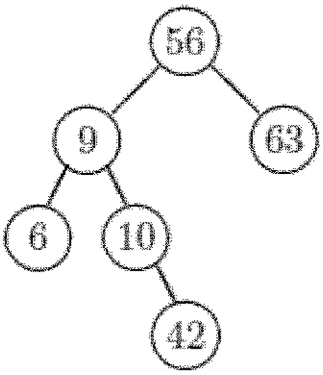
Question B4

(a) Given the following min heap, draw the min heap that would result after deleting the minimum element.



(4 marks)

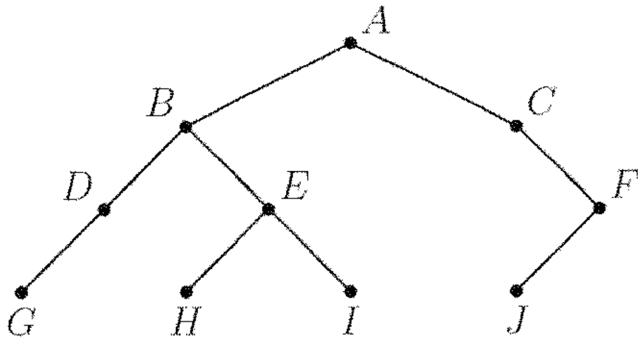
(b) Given the following tree that was obtained by inserting the element 42 into an AVL tree. The tree no longer satisfies the AVL invariant, but the invariant can be re-established by performing two rotate operations. Show the result after each rotation.



(6 marks)

Question B5

(a) Given the following binary tree:



(i) List the sequence of nodes visited by preorder traversal. (2 marks)

(ii) List the sequence of nodes visited by inorder traversal. (2 marks)

(iii) List the sequence of nodes visited by postorder traversal. (2 marks)

(b) Write the pseudocode of a preorder traversal of a binary tree. (4 marks)

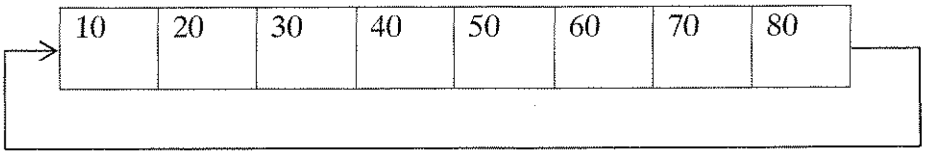
- End of Section B -

Section C (20%) – Long Questions

Answer any ONE of the TWO questions in this section in the answer book provided. Each question carries 20 marks. If more than ONE question is answered, only the first ONE question answered will be marked.

Question C1

In a standard queue data structure, re-buffering problem occurs for each dequeue operation. We can solve this problem by joining the front and rear ends of a queue to make the queue as a *Circular Queue* as the structure shown below. Circular queue is a linear data structure. It follows FIFO principle and the last node is connected back to the first node to make a circle.



10	20	30	40	50	60	70	80

(a) Write a pseudocode for *Enqueue* operation of a circular queue. (5 marks)

(b) Write a pseudocode for *Dequeue* operation of a circular queue. (5 marks)

(c) You have a circular queue with a capacity of 8 elements. Show the contents of the array after the following sequence of enqueue and dequeue operations, and where the front and back indices will point. If an array entry is empty, either because it has never been filled, or because its value has been dequeued, then leave its box blank.

```
enqueue(6);
enqueue(5);
enqueue(3);
enqueue(2);
dequeue();
dequeue();
```

(5 marks)

(d) Now you receive the following additional set of enqeueue and dequeue operations. Show the contents of the array and where the front and back pointers point after these additional operations have been applied to the queue.

```
enqueue(7);
enqueue(8);
enqueue(2);
enqueue(11);
dequeue();
dequeue();
enqueue(9);
```

(5 marks)

Question C2

Given the following Java code:

```
public class QuestionCTest {
  public static void show(int [] arr) {
         for (int i = 0; i < arr.length-1; i++)
              System.out.print(arr[i] + ", ");
         System.out.println(arr[arr.length-1]);
  }

  public static void sorting(int [] element) {
     // fill in your code here in part (b)
  }

  public static void main(String [] args) {
   int []  number = {7, 52, 23, 75, 16, 37, 2, 46, 14 };
           sorting(number);
  }
}
```

The output of the above program being executed is:

```
7, 23, 52, 16, 37, 2, 46, 14, 75
7, 23, 16, 37, 2, 46, 14, 52, 75
7, 16, 23, 2, 37, 14, 46, 52, 75
7, 16, 2, 23, 14, 37, 46, 52, 75
7, 2, 16, 14, 23, 37, 46, 52, 75
2, 7, 14, 16, 23, 37, 46, 52, 75
2, 7, 14, 16, 23, 37, 46, 52, 75
2, 7, 14, 16, 23, 37, 46, 52, 75
```

(a) Specify the name of the sorting method carried out by `sorting()` method according to the given output. (2 marks)

(b) Complete the `sorting()` method so that it gives the above output. (6 marks)

(c) A better algorithm for `sorting()` method is that it can terminate the sorting process earlier once it has found the elements in the array are in ascending order. Modify the `sorting()` method to this early terminating version. (4 marks)

(d) Show the output after the early terminating version in **part (c)** is executed. (2 marks)

(e) Modify your code in **part (c)** so that the `sorting()` method sorts the given number in descending order. (2 marks)

(f) You have to sort a list *L* consisting of a sorted list followed by a few "random" elements. Which sorting method would be especially suitable for such a task? Explain your answer. (4 marks)

- End of Section C -

- END OF PAPER -

P